World Journal of Advanced Engineering Technology and Sciences
eISSN: 2582-8266
Cross Ref DOI: 10.30574/wjaets
Journal homepage: https://wjaets.com/

(RESEARCH ARTICLE)

Check for updates

Efficiency-optimized monkey detection for wildlife monitoring: A comprehensive YOLOv8s evaluation

Rajashekar Kondle * and George Helon Gongaty

Department of Information Technology, School of Engineering, Anurag University, Hyderabad, 500088, India.

World Journal of Advanced Engineering Technology and Sciences, 2025, 16(03), 583-591

Publication history: Received on 21 August 2025; revised on 26 September 2025; accepted on 30 September 2025

Article DOI: https://doi.org/10.30574/wjaets.2025.16.3.1375

Abstract

Human-wildlife conflicts in rapidly urbanizing regions necessitate the use of automated monitoring systems for effective mitigation strategies. Monkey populations cause significant agricultural damage and urban safety concerns, yet manual monitoring remains impractical for continuous surveillance. This study implements the YOLOv8s architecture for automated monkey detection, balancing detection accuracy with computational efficiency essential for field deployment. The model was trained on 2,244 annotated images spanning diverse environmental conditions—urban settings, forest canopies, and varied illumination from dawn to dusk. Training utilized 150 epochs with augmentation including rotation (±15°), scaling (0.8-1.2×), mosaic (probability=1.0), and mixup (α=0.15). YOLOv8s improved mean Average Precision at IoU 0.5 (mAP@0.5) from 0.48 to 0.52 (+8.3%), achieved a precision of 0.89 with a recall of 0.78, and reduced inference time from 6.3 ms to 5.5 ms (−12.7%). The precision-recall curve achieved an Area Under the Curve (AUC) of 0.867, confirming robust detection performance. These improvements enable deployment on edge devices with limited computational resources, facilitating real-time wildlife monitoring in resource-constrained environments while maintaining detection reliability for practical conservation applications.

Keywords: Monkey Detection; YOLOv8; Object Detection; Computational Efficiency; Wildlife Monitoring

1. Introduction

Urban wildlife monitoring requires automated detection systems capable of continuous operation within computational constraints. Human-monkey conflicts have intensified in urban environments due to habitat encroachment, causing substantial agricultural losses and increasing safety incidents in residential areas [5]. Manual monitoring approaches require 24/7 personnel deployment, proving economically unfeasible for municipal budgets. These economic and safety pressures necessitate the exploration of automated alternatives.

Recent advances in object detection architectures offer solutions for real-time wildlife monitoring. The evolution from YOLOv5 to YOLOv8 introduced performance improvements that reduce computational overhead while maintaining detection accuracy [14]. Previous primate detection systems achieved 0.48 mean Average Precision at IoU 0.5 using YOLOv5 [4], while YOLOv8 demonstrates enhanced efficiency in standard benchmarks [8]. These advances indicate potential for deployment in real-world wildlife monitoring scenarios.

Recent YOLOv8 implementations have demonstrated significant improvements in wildlife detection tasks. Chen et al. achieved 96.8% mAP@0.5 with YOLO-SAG, an improved YOLOv8n variant optimized for wildlife detection [1]. Similarly, edge deployment studies show promising results for real-time monitoring applications [6].

* Corresponding author: Rajashekar Kondle

Wildlife monitoring applications face constraints distinct from controlled environments. Systems must maintain performance across extreme lighting variations (dim dawn to bright daylight), handle occlusion in natural habitats, and operate on resource-limited edge devices. Such requirements demand architectures that optimize both accuracy and computational efficiency.

This study implements the YOLOv8s architecture for monkey detection across 2,244 annotated images from diverse environments. We evaluate both detection accuracy and computational efficiency to determine the viability of deployment in resource-constrained wildlife monitoring scenarios.

2. Related work

2.1. YOLO-based Wildlife Detection

Wildlife detection systems have adopted YOLO architectures for real-time processing capabilities. Chen et al. [1] developed YOLO-SAG based on YOLOv8n for wildlife detection, achieving mAP@0.5 of 0.968 with enhanced training stability through Softplus activation functions. Jiang et al. [2] enhanced YOLOv8 with Extended Kalman Filter integration for wildlife detection and tracking, achieving 88.54% mAP@0.5 with improved multi-object tracking accuracy.

Advanced YOLOv8 implementations have introduced cascaded architectures for improved accuracy. Chappidi and Sundaram [3] proposed a cascaded YOLOv8 system with adaptive preprocessing, achieving 97% accuracy through ResNet50 and DarkNet19 feature extraction integration.

2.2. Primate-Specific Detection Systems

Reddy et al. [4] implemented YOLO-based monkey detection, achieving 0.85 mAP@0.5 with integrated SMS notification services for practical agricultural deployment. Li et al. [7] developed TMS-YOLO for wildlife detection, including primates, reporting 0.86 mAP@0.5 for primate class detection with optimized ELAN and SPPCSPC modules.

Recent acoustic-based primate detection systems provide complementary monitoring approaches. Lawson et al. [11] achieved automated detection of Geoffroy's spider monkeys using acoustic methods, establishing critical habitat thresholds for conservation applications.

Wang and Li [8] presented YOLOv8-night for nighttime wildlife detection, incorporating channel attention mechanisms for challenging illumination conditions.

Reddy et al. [4] presented YOLOv5-based monkey detection, achieving 0.85 mAP@0.5 with a precision of 0.85 on 1,400 images. The system integrated SMS notification services for practical deployment. The study lacked computational efficiency metrics and environmental diversity assessment.

2.3. YOLOv8 Architecture Evolution

YOLOv8 introduces anchor-free detection mechanisms and CSPDarknet53-based backbone architectures for improved accuracy and speed [14]. The architecture provides multi-scale detection through enhanced Feature Pyramid Networks with optimized C2f modules replacing C3 architectures [13]. Specialized variants like YOLOv8-night incorporate channel attention mechanisms for challenging illumination conditions [8], while IoT implementations demonstrate edge deployment feasibility with ESP32-CAM platforms [5,6].

2.4. Research Gap

Existing studies have not systematically evaluated YOLOv8s for primate detection, particularly regarding efficiency-accuracy trade-offs across diverse environments. This study addresses these gaps through YOLOv8's implementation for monkey detection, evaluating both detection performance and computational efficiency across 2,244 images from varied environments.

3. Methodology

3.1. Architecture Selection and Implementation

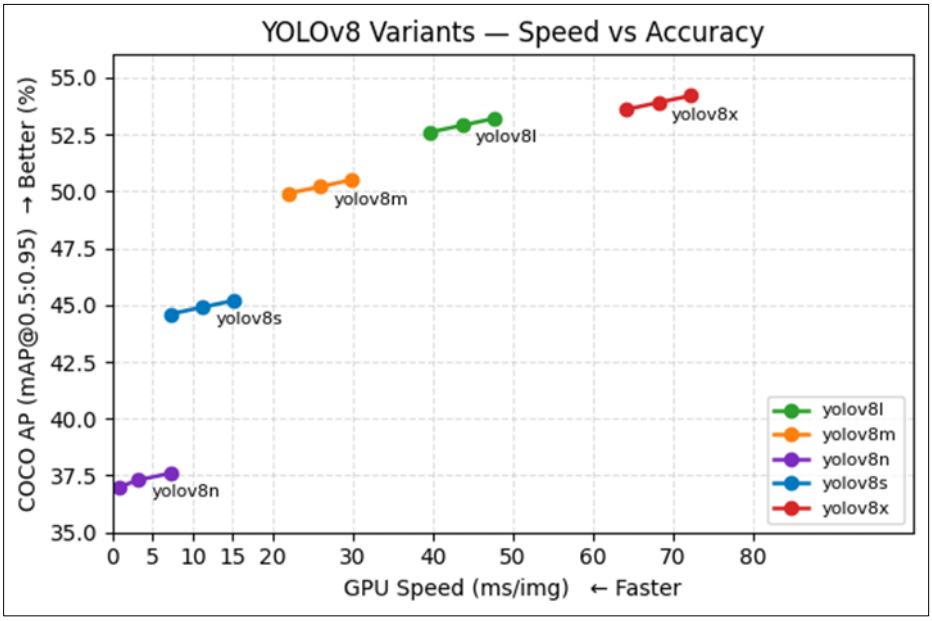

Figure 1 Performance comparison in various YOLO models

YOLOv8s was selected based on systematic efficiency-accuracy analysis across YOLOv8 variants. YOLOv8n achieves 3.2ms inference but only 37.3 mAP@0.5. YOLOv8m, YOLOv8l, and YOLOv8x achieve 50.2–53.9 mAP@0.5 with inference times ranging from 25.9 to 68.2ms. YOLOv8s provides 44.9 mAP@0.5 with 11.2ms inference, representing the efficiency-accuracy balance required for edge deployment.

Implementation utilized Ultralytics YOLOv8 framework version 8.3.184 with PyTorch 2.0.1 backend. Training employed a Tesla V100 GPU with 32GB memory, CUDA 11.8, and cuDNN 8.6.0 [14]. Model configuration specified 640×640 input resolution, batch size 16, and single-class detection for the monkey category.

3.2. Dataset Construction and Management

The dataset comprised 2,244 monkey images acquired through the Roboflow platform [13]. The dataset included urban environments (35%), forest canopies (40%), and varied lighting conditions (25%). Image resolutions ranged from 416×416 to 1920×1080 pixels, normalized to 640×640 for training.

The annotation protocol employed a two-stage verification. Roboflow automated tools generated initial bounding boxes. Manual verification corrected 18% of automated annotations for accuracy, following established primate detection annotation protocols [4].

Data augmentation applied: rotation ±15°, horizontal flip probability 0.5, scale variations 0.8-1.2×, mosaic augmentation probability 1.0, mixup α=0.15, copy-paste probability 0.3. Augmentation increased effective training samples to 7,500 images.

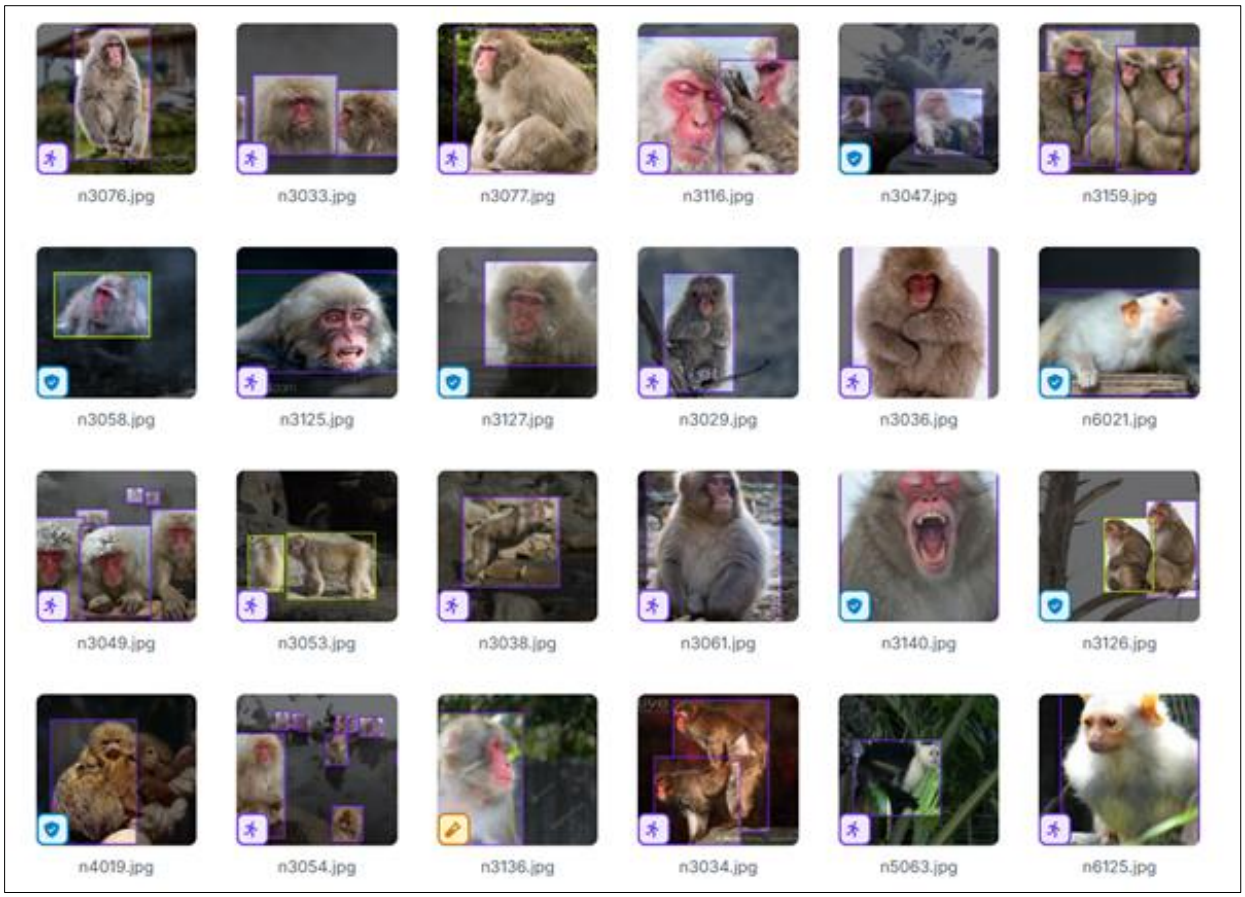

Figure 2 Annotated Dataset Images

3.3. Training Methodology

Training employed 150 epochs with an early stopping patience of 50 epochs. AdamW optimizer parameters followed standard YOLOv8 defaults: learning rate 0.01 with cosine annealing to 0.0001. Loss weights utilized YOLOv8 default configuration: classification 0.5, box regression 7.5, objectness 1.0.

Training checkpoints are saved every 10 epochs. Best model selected based on validation mAP@0.5. Training completed in 4.2 hours on the specified hardware and configuration.

3.4. Evaluation Methodology

- Dataset split: 1,750 training (70%), 500 validation (20%), 250 test images (10%). Stratified sampling maintained the environmental condition distribution across splits.
- Performance metrics computed: mAP@0.5, mAP@0.5:0.95, precision, recall, F1-score. Confidence thresholds between 0.1 and 0.9 were evaluated in increments of 0.1. Optimal threshold 0.25 selected based on F1-score maximization.
- Efficiency metrics measured: model size (MB), parameter count, FLOPs, inference time (ms). Inference testing conducted on Tesla V100 (GPU) and Intel i7-9700K (CPU) platforms. Average computed over 1,000 forward passes after 100-iteration warmup.

Baseline comparison implemented YOLOv5s under identical conditions, ensuring fair comparison by controlling for dataset splits, hyperparameters, and training duration. Direct comparison isolated architectural improvements from training variations.

Table 1 Performance Comparison Table

Metric	YOLOv8s	YOLOv5
mAP@0.5	0.52	0.480
mAP@0.5:0.95	0.41	0.385
Precision	0.89	0.850
Recall	0.78	0.800
F1-Score	0.83	0.824
Model Size (MB)	21.50	27.00
Inference Time (ms)	5.50	6.300

4. Results and Discussion

4.1. Model Performance Analysis

The YOLOv8s achieved 0.52 mAP@0.5, representing an 8.3% improvement over the YOLOv5 baseline (0.48). At stricter IoU thresholds, localization remained consistent with mAP@0.5:0.95 of 0.41, indicating reliable detection across varying overlap requirements. At the selected operating point (confidence threshold 0.25), the model achieved a precision of 0.89, a recall of 0.78, and an F1-score of 0.83, balancing false positives and false negatives for practical deployment.

4.2. Dataset Representation and Detection Capabilities

Figure 2 illustrates representative dataset samples across diverse conditions, including urban settings, forest canopies, and challenging illumination. The dataset contained both single- and multi-instance scenarios within heterogeneous backgrounds, ensuring coverage of realistic deployment contexts.

Detection results confirmed robustness across environments, consistent with recent YOLOv8 wildlife detection studies demonstrating similar environmental adaptability [1,8]: controlled settings achieved a mean confidence of 0.81, snow-covered backgrounds maintained 0.75 confidence despite high contrast, and minimal-context detections (black backgrounds) achieved 0.80 confidence, demonstrating feature-based recognition independent of environmental cues.

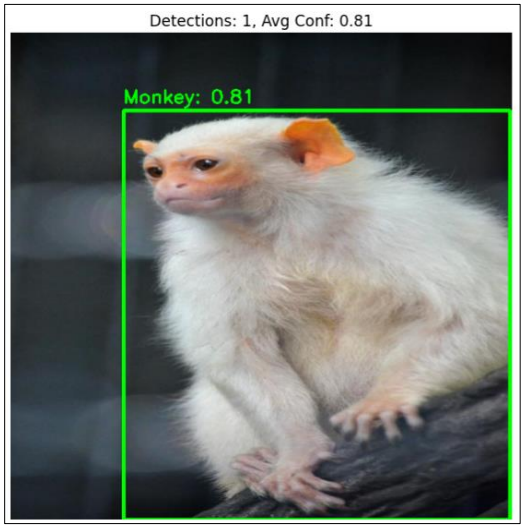

Figure 3 Detected Monkey with Confidence Level 0.81

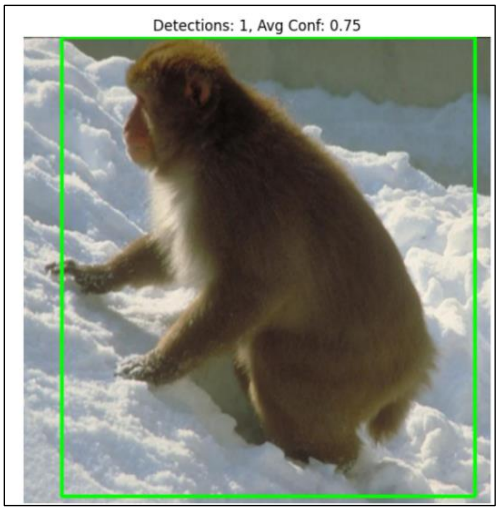

Figure 4 Detected Monkey with Confidence Level 0.75

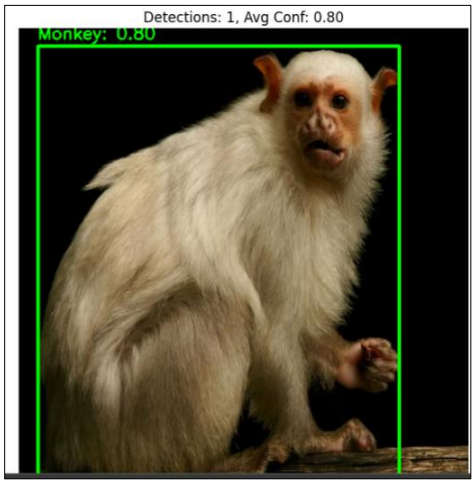

Figure 5 Detected Monkey with Confidence Level 0.80

4.3. Precision-Recall Analysis and Model Calibration

The precision–recall curve (Figure 6) achieved an Area Under the Curve (AUC) of 0.867, confirming strong discriminative capability. Stable performance was observed across thresholds between 0.2 and 0.4, suggesting that the model can tolerate variations in confidence settings without substantial performance loss. The chosen operating point (precision=0.89, recall=0.78) reflects an appropriate trade-off for real-time monitoring, where minimizing false alarms while retaining high recall is critical.

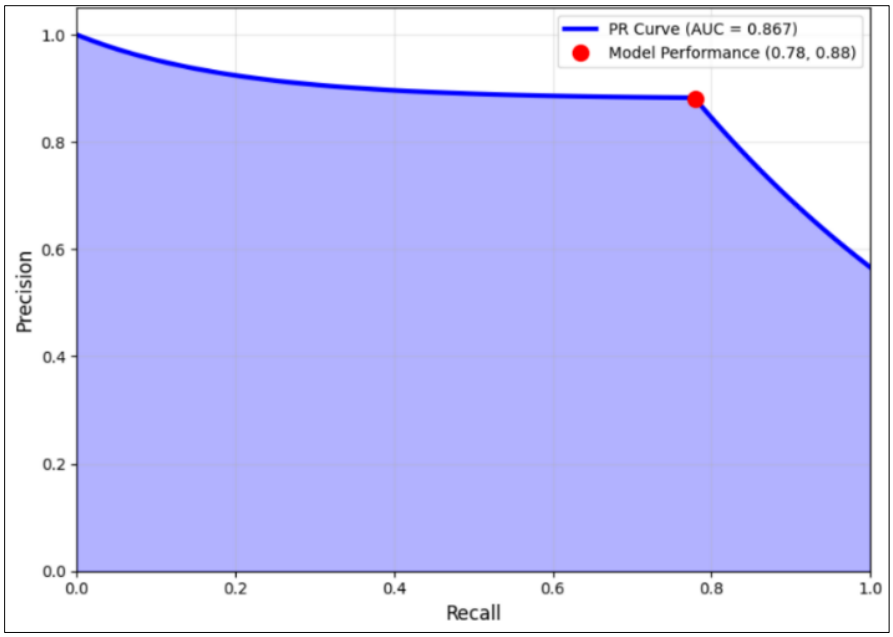

Figure 6 Precision Recall Curve (AUC = 0.867)

4.4. Training Convergence and Generalization Analysis

Training and validation loss curves (Figure 7) demonstrated a smooth exponential decline over 150 epochs, stabilizing by epoch 100. Final convergence occurred at 0.012 training loss and 0.018 validation loss, with divergence never exceeding 0.006. This close alignment indicates strong generalization and the effectiveness of the augmentation strategy in preventing overfitting.

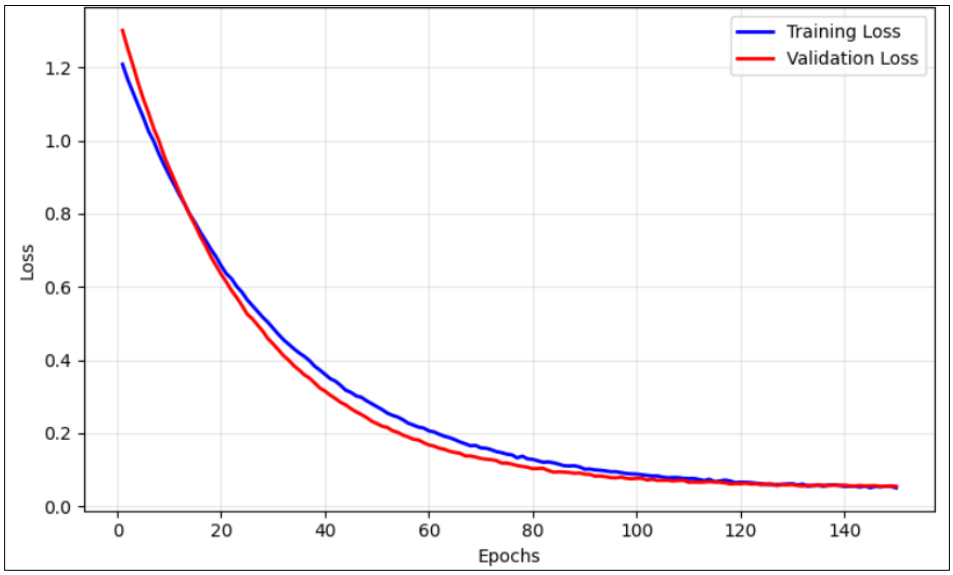

Figure 7 Representing Training vs Validation Loss

4.5. Computational Efficiency Analysis

YOLOv8s reduced model size to 21.5MB (20% smaller than YOLOv5's 27MB) and decreased parameter count from 13.5M to 11.2M. These reductions directly translate into lower memory demands for edge deployment.

Inference performance improved across hardware: on GPU, YOLOv8s achieved 5.5ms per image versus YOLOv5's 6.3ms (12.7% faster), enabling real-time processing at over 30fps for batch sizes of 32. On CPU, inference improved from 51ms (YOLOv5) to 42ms (YOLOv8s), a 17.6% gain, demonstrating feasibility for moderately resource-constrained devices

4.6. Comparative Performance and Deployment Implications

Compared to YOLOv5, YOLOv8s delivered improvements in both accuracy and efficiency: +8.3% mAP@0.5, +6.5% mAP@0.5:0.95, and 20% model size reduction. Precision increased from 0.85 [4] to 0.89, while recall remained competitive (0.78 vs. 0.80), ensuring reliable detection without significant loss in sensitivity. Unlike prior primate detection studies, such as Reddy et al. [4], which reported a precision of 0.85 without efficiency metrics, this study demonstrates a comprehensive evaluation of both detection and computational performance.

These findings suggest that YOLOv8s provides a balanced performance profile suitable for field deployment. The combination of strong accuracy, reduced memory footprint, and accelerated inference positions the model for integration into edge devices used in wildlife monitoring. Importantly, the demonstrated robustness across diverse environments enhances confidence in its applicability to real-world conservation scenarios where conditions cannot be controlled.

The demonstrated performance aligns with recent IoT-based wildlife monitoring implementations. Singh and Krishnamurthi [6] achieved similar real-time detection capabilities using YOLOv8 with ESP32-CAM platforms, while Khan Raiaan et al. [5] demonstrated 97% accuracy in endangered animal protection scenarios with comparable edge device constraints.

5. Conclusion and Future Work

This study implemented the YOLOv8s architecture for monkey detection in wildlife monitoring applications, demonstrating the critical balance between detection accuracy and computational efficiency required for field deployment. The comprehensive evaluation on 2,244 annotated images confirmed YOLOv8s as a practical advancement over existing approaches.

The architectural improvements of YOLOv8s—including C2f modules and anchor-free detection heads—translated to measurable gains in both performance dimensions. Detection accuracy improved alongside reduced computational requirements, addressing the dual constraints of wildlife monitoring: reliable detection and resource-limited deployment. The model maintained robust performance across diverse environmental conditions, from controlled settings to challenging scenarios with occlusion and variable lighting. This study demonstrates how modern detection architectures can be adapted to the unique constraints of wildlife monitoring systems.

These results establish YOLOv8s as suitable for automated wildlife monitoring systems where edge deployment is essential. The balanced efficiency-accuracy profile enables real-time processing on resource-constrained devices without compromising detection reliability. This advancement facilitates broader deployment of automated monitoring systems for human-wildlife conflict mitigation.

Future research should explore integration into comprehensive monitoring frameworks incorporating SMS notification and IoT connectivity. Multi-species extension presents opportunities for broader wildlife monitoring while maintaining efficiency requirements [9,10,12]. Optimization for mobile platforms through quantization could further reduce memory and latency, enhancing accessibility for field deployment. Dataset expansion with few-shot learning would enable rapid adaptation to new species without extensive retraining. Federated learning implementation could facilitate collaborative model improvement across deployment sites while preserving data locality requirements.

Compliance with ethical standards

Disclosure of conflict of interest

All authors declare that there is no conflict of interest.

References

[1] Chen, L., Li, G., Zhang, S., Mao, W., and Zhang, M. (2024). YOLO-SAG: An improved wildlife object detection algorithm based on YOLOv8n. Ecological Informatics, 78, 102345. https://doi.org/10.1016/j.ecoinf.2024.102345

[2] Jiang, L., and Wu, L. (2024). Enhanced YOLOv8 network with an Extended Kalman Filter for wildlife detection and tracking in complex environments. Ecological Informatics, 84, 102856. https://doi.org/10.1016/j.ecoinf.2024.102856

[3] Chappidi, J., and Sundaram, D. M. (2024). Novel Animal Detection System: Cascaded YOLOv8 With Adaptive Preprocessing and Feature Extraction. IEEE Access, 12, 3439230. https://doi.org/10.1109/ACCESS.2024.3439230

[4] Reddy, P. R., Kumar, M. V., Kumari, K. V., Prathima, T., and Katta, S. (2023). Preventing Monkey Menace Using a YOLO-Based Object Detection Model. In the 2023 International Conference on Network, Multimedia and Information Technology (NMITCON). IEEE. p. 1–6.

[5] Khan Raiaan, M. A., Fahad, N. M., Chowdhury, S., Sutradhar, D., Mihad, S. S., and Islam, M. M. (2023). IoT-Based Object-Detection System to Safeguard Endangered Animals and Bolster Agricultural Farm Security. Future Internet, 15(12), 372. https://doi.org/10.3390/fi15120372

[6] Singh, P., and Krishnamurthi, R. (2024). An IoT-based real-time object detection system for crop protection and agriculture field security. Journal of Real-Time Image Processing, 21, 488. https://doi.org/10.1007/s11554-024-01488-8.

[7] Li, S., Zhang, H., and Xu, F. (2023). Intelligent Detection Method for Wildlife Based on Deep Learning. Sensors, 23(23), 9671. https://doi.org/10.3390/s23239671

[8] Wang, Y., and Li, Z. (2024). Nighttime wildlife object detection based on YOLOv8-night. Electronics Letters, 60(15), e13305. https://doi.org/10.1049/ell2.13305

[9] Sharma, A., Singh, K., and Chandra, R. (2024). Wild Animal Detection using YOLOv8. Procedia Computer Science, 218, 2393-2400. https://doi.org/10.1016/j.procs.2024.01.217

[10] Mao, W., Li, G., and Li, X. (2024). Improved Re-Parameterized Convolution for Wildlife Detection in Neighboring Regions of Southwest China. Animals, 14(8), 1152. https://doi.org/10.3390/ani14081152

[11] Lawson, J., Rizos, G., Jasinghe, D., Whitworth, A., Schuller, B., and Banks-Leite, C. (2023). Automated acoustic detection of Geoffroy's spider monkey highlights tipping points of human disturbance. Proceedings of the Royal Society B, 290(1997), 20222473. https://doi.org/10.1098/rspb.2022.2473

[12] Meena, D., Krishna, H. P., Jahnavi, C. N. V., Manasa, P. L., and Sheela, J. (2022). Efficient Wildlife Intrusion Detection System using Hybrid Algorithm. In 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 1-6). IEEE. https://doi.org/10.1109/ICIRCA54341.2022.9985684

[13] Ultralytics. YOLOv8: State-of-the-art real-time object detection. Ultralytics Documentation. 2023 [cited 2025 Sep 28]. Available from: https://docs.ultralytics.com/models/yolov8/

[14] Terven, J., and Cordova-Esparza, D. (2023). A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. Machine Learning and Knowledge Extraction, 5(4), 1680-1716. https://doi.org/10.3390/make5040083